MC48 Intra-term Exam 2

Serial #:

This exam is an open-book, open-notes, individual-work exam. Please write your name only on this page. Do not turn the page until instructed, in order that everyone may have the same time. Then, be sure to look at all problems before deciding which one to do first. Some problems are easier than others, so plan your time accordingly. You have 50 minutes to work.

Write the answer to each problem on the page on which that problem appears. You may use the back of that same sheet if you need additional space. You may also request additional paper.

Name: ______________________________

Problem	Page	Possible	Score
1	2	12	
2	3	12	
3	4	12	
Total		36	

1. [**12 Points**] A new instruction, `beqir` is to be added to the MIPS instruction set. It is an I-format instruction, that is, it has the following general layout:

```
bits: 31-26   25-21   20-16    15-0
     +------+-------+-------+--------------------+
     |beqir | Rs    | Rt    |  immediate         |
     +------+-------+-------+--------------------+
```

The meaning of this instruction is as follows. First the contents of the register specified by Rs is compared with the sign-extended immediate. If they are unequal, then nothing happens (execution falls through to the next instruction). However, if they are equal, then a branch occurs. The contents of the register specified by Rt tells how far (forwards or backwards) the branch should go; it is added to the address of the instruction following the `beqir` instruction in order to determine the address of the instruction that should be executed next. That is, if the address of the `beqir` is PC, and the value stored in the Rt register is x, then when the branch is taken, the next instruction fetched should be from address $PC + 4 + x$.

(a) This instruction is to be added to the multi-cycle implementation. The datapath shown in figure 5.35 is adequate to execute this instruction without any change. (Be sure to use the version of this figure from the errata.) The instruction can be processed in four clock cycles, of which the first two are the same as any other instruction. Specify what should be done in the third and fourth clock cycle, using the C-like notation used in Figure 5.34 on page 365 (this notation is also used on page 364).

(b) Now show what two new states would need to be added to the finite state machine control, which is figure 5.43. (Again, be sure to use the version of this figure that is in the errata.)

2. [**12 Points**] The following instructions are executed by the pipelined MIPS implementation in the order given:

```
add $t0, $t1, $t2    # (instruction 1)
add $t1, $t2, $t3    # (instruction 2)
add $t3, $t4, $t5    # (instruction 3)
add $t5, $t0, $t1    # (instruction 4)
add $t3, $t3, $t5    # (instruction 5)
lw  $t4, 16($t5)     # (instruction 6)
add $t0, $t4, $t3    # (instruction 7)
```

Answer the following questions about this instruction sequence. Assume that forwarding has been introduced as we discussed to minimize the need for stalls. This includes internal forwarding within the register bank, such that if a given register is both written and read in the same clock cycle, the value being written will also be read.

(a) How many stalls are needed? Between what instructions should the stall bubble(s) be introduced?

(b) At the time when instruction 7 is in the EX pipeline stage, what is in the MEM and WB pipeline stages?

(c) Fill in the following table so that it shows which pipeline register each ALU input comes from when each of the following instructions is in the EX stage. Each entry you put in the table should be ID/EX, EX/MEM, or MEM/WB:

instruction	**Top ALU input**	**Bottom ALU input**
3		
4		
5		
6		
7		

3. [**12 Points**] In your most recent homework, you calculated that the pipeline registers (including control) had 64, 147, 107, and 71 bits. Taken together with the 32-bit PC, this means that the pipelined implementation has a total of 421 bits of state, not counting the instruction or data memories or the 32 32-bit registers in the main register bank. How many bits of state does each of the other two implementations have, again excluding the memory(s) and the main register bank? (The single-cycle implementation is shown in figure 5.29 on the front cover of the errata; the multiple-cycle implementation is shown in figure 5.35 in the errata.) Be sure to include any state that is involved in the control (just as the above pipeline register sizes included control). You must justify your answers by enumerating what the state consists of.